A Survey on Dynamic Honeypots

Hamid Mohammadzadeh.e.n, Roza Honarbakhsh, and Omar Zakaria

***Abstract*—There are a lot of critical issues in the network security concern. One of these issues is deployment of dynamic honeypot. To overcome this matter, plug and play honeypot methods are being promoted by some researchers. Therefore, the main aim of this paper is to study on the usability and security features of the plug and play concept in order to have easy deployment of dynamic honeypot on a network environment. It also offers type of integration of existing network tools to deploy dynamic honeypot based on its placement within network topology.**

***Index Terms*—Dynamic honeypots, honeynet, plug and play honeypot.**

I. Introduction

Today's it is obvious that the number of vulnerable threat of network assets increase significantly. The major examples of such threats are worms which can be found in the Internet. In spite of decades of research and experience, still incapable to make secure computer systems and also they immobilized manufacture ways to measure a computer systems and networks level of security. This is certainly true that intrusion to a network or system can never be eliminated but security helps reduce risk to an organization and its information-related resources.

However, three particular concepts surrounded the IT security includes *Detection*, *Prevention* and *Reaction***.** In one hand, you can't find end-to-end equipments solution to cover all or even two of concepts. In the other hand, secure digital resources in an organization by ruling detect and prevent vulnerabilities before exploited. Looking at network intrusion detection systems (NIDS) before starting with honeypots, could be useful to motivate the use of honeypots. NIDS by providing the amount of functional information causes an increasing number of protocols that employ encryption to protect network traffic from eavesdroppers and decreasing in the face of ever more complicated evasion techniques. Effectiveness of NIDSs is weak because of high false positive rates. Using honeypots assists to decrease the number of the false positives. In network security an Intrusion Prevention System (IPS) is the method of that monitors network and/or system activities for malicious or unwilling behavior. Moreover, IPS is accountable attempting to stop discovered possible occurrences in real-time.

In the other hand intrusion prevention systems are the devices of performing intrusion detection [1]. A novel strategy in the modernized defensive network security is *Deception*. In fact, this issue is relatively in intrusion detection field. The idea which follows these systems is to provide systems or services that deceive the intruder. Therefore, these systems help in learning the techniques that invaders use and they can also be viewed as a decoy to amuse hackers from the real systems and services. Thus, *Honeypots* can be categorized as deception systems. The purpose of honeypot in security community refers to a computer system that is used as a trap to catch the attention of attackers to attack this computer system. If so, gathering useful information from attacker, firstly, allows us to present controlled information to our adversaries. Secondly, a honeypot allows us to collect information concerning our attackers. Finally, a honeypot can serves as a disincentive against future attacks .This information could be pattern of attack, unknown security holes, tools and techniques they used, and even the keystrokes of attacker. Similarly, this information can be used to prevent, block, or turn off the attack. In additional, based on this information enforcing and strengthen existing intrusion detection tools or network firewalls will be possible. It is considerable that honeypots should not be viewed as a solution to network security; they should be seen as an aid to it [2].

Regardless to detection and prevention the values of a honeypot is directly comparative to the amount and type of information we can successfully obtain from it. Sideways from information gathering, a honeypot has the capabilities of distracting adversaries from more valuable machines on a network, and can provide early warning signs concerning a new type of attack or exploitation trends, and allows in-depth examination of adversaries during or after exploitation of a host [3].

II. Background

Honeypots in contrast to other security means like IDSs or firewalls does not concentrate on a specific problem; instead it can contribute as a tool with other parts of overall security architecture. We are using honeypot as an appliance to accumulate evidence, information and also to gain as much acquaintance as possible, especially on the attack patterns. Generally honeypots can be classified into three main categories:

A. Classification of Honeypots :Interaction Level

1) Low-Interactions

On low-interaction honeypots, tools are installed due to imitate operating systems and services and then interact with the attackers and malicious code. Moreover, this type of honeypot has a tiny chance of being compromised and is ideal for production networks. Honeypots work entirely

Manuscript received January 25, 2012; revised March 3, 2012.

H. Mohammadzadeh.e.n and O. Zakaria were with the faculty of Computer Science and Technology, University of Malaya, Kuala Lumpur, 50603 Malaysia (e-mail: h.network@ gmail.com ; omar@upnm.edu.my).

S. B. Roza Honarbakhsh was with University Technology Malayisa, Kuala Lumpur, Malaysia (e-mail: roza.honarbakhsh@ gmail.com).

imitating operating systems and services. The attacker's activities are limited to the honeypot's level and quality of emulation [4].However, the amount of information that provided by these simple honeypots are limited. Information about transactions and some of the attacker's activities are obtained by monitoring the low-interaction honeypots. Low-interaction honeypots are not appropriate for discovering or interacting with unknown and unexpected behaviors or attacks. For those who have never worked before with Honeypots, this kind of honeypot is recommended,"Fig. 1".

2) High-Interaction

In classification of honey pots regarding to their level of interaction the final and most advanced of honeypots are the high-interaction honeypots. These kinds of honeypots are really time consume to design, therefore difficult to manage and maintain. It has been clearly enunciated that as the level of interaction of honeypots increases, among the three types of honeypots, this honeypot has more risks and complexity because they involve real operating systems with real applications but the information and evidence gathered for analysis are bountiful and rewards also increases,"Fig. 2".

3) Medium-Interaction

Another type of honeypots which tried to combine the benefits of both approaches of low and high interaction is the medium-interaction Honeypots. This is slightly more advanced than low-interaction honeypots, but a little less intricate than high-interaction honeypots. No real operating system is provided, but application layer virtualization. These kinds of honeypots do not aim at fully simulating a fully operational system environment, nor do they implement all details of an application protocol. All that these kinds of honeypots does to provide adequate responses that known exploits await on certain ports that will trick them into sending their payload [5]. To ensure that attackers avoid damaging other systems, we have to developed mechanisms.This increased functionality is not vulnerable to exploitation by an attacker.

B. Placement of Honeypots

1) Behind of the Firewall

The vast majority of honeypots are installed inside the firewalls"Fig3". So that they can be better controlled, although it is possible to install them outside firewalls .Installing honeypot inside firewall will increase the risks in internal network, particularly if the internal network was not secured against the honeypots

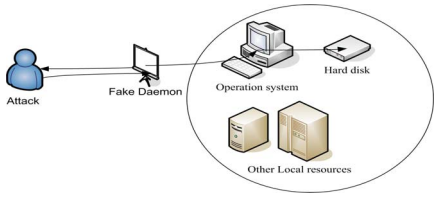

Fig. 1. Low interaction honeypot.

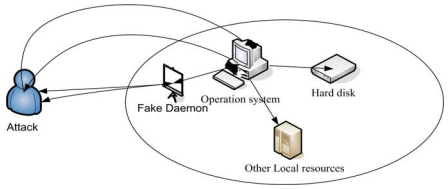

Fig. 2. High interaction honeypot.

By additional firewall. Similarly, IDS signature can't generate alarm every time the honeypot is attacked or scanned. The greatest problem occurs when the internal honeypot is compromised by external attacker. As a result, this traffic will be unstopped as honeypot traffic; thereby, the attacker has the access to the internal network through the honeypot. In fact, the main reason to placed honeypot inside firewall is to analyze the internal attackers. In additional, with internal honeypot it is also possible to find a miss-configuration of firewall [6].

2) In Front of the Firewall

However, utilize compromised system in front of firewall reduces the internal network risks"Fig4". A role that firewall in a honeypot plays, is in the opposite way that a normal firewall works. Indeed , the honeypot firewall restricts what the system send back out and allows all traffic to come in from the internet ,instead limits what comes into a system from the Internet. By tempting a hacker into a system, a honeypot serves several purposes: the administrator can observes the hacker utilize the vulnerabilities of the system; therefore, understanding the weaknesses can assest to redesign the system to approach our purpose. As a result, the hacker can be trapped and blocked while annoying to obtain root access to the system. This information will help designers to create more secure system that is possibly impenetrable to future hackers. The main advantages of placing honeypot in front of firewall is not restricted by any rule of firewall or IDS .In the other hand the disadvantage is that the internal attackers can't be trap that easy, specifically when firewall limits the outbound traffics.

C. Honeypots Classification Based on IP Address

1) Physical Honeypot:

It is a real machine with its own IP address. Potentially physical honeypots are often high-interaction because they are expensive resource to develop and maintain. Likewise, the system can be entirely compromised. To illustrate, if you want to execute physical honeypots for a particular range of IPs on your LAN, you would have to build a separate occurrence of a honeypot for each physical IP address. As a result, deploying a physical honeypot is time consuming and costly due to different operating systems entail specialized hardware; therefore ,every honeypot requires its own physical system.

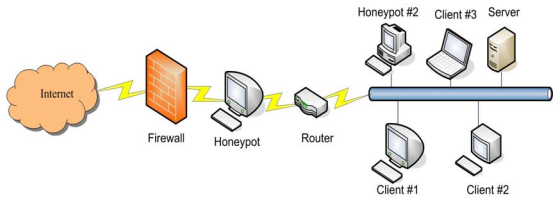

Fig. 3. Honeypot behind of the firewall.

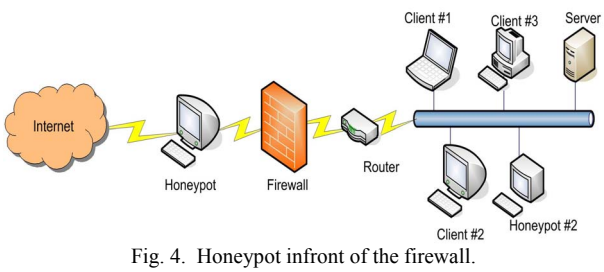

Fig. 4. Honeypot infront of the firewall.

2) Virtual Honeypot

It is simulated by another machine that responds to network traffic. However, virtual honeypots usually categorize low interaction due to low implementation and maintenance costs. A virtual honeypot can simulate multiple Operating Systems, services and a separate TCP/IP stack for each instance of a honeypot on that one machine.

The comparison between virtual and physical honeynets indicates that both virtual and real machines collect roughly the same amount of data .The most essential difference between the virtual and real honeypots is the maintenance aspect of both systems. Meanwhile, setting up the honeynet proved to be easier on physical systems since high-quality knowledge of networking is necessary for deploying virtual honeynets, maintenance of virtual honeynets proved to be much simpler than the real system. If a honeypot is compromised, then it is necessary to reverse the data to the previous state. This process is much straightforward in a virtual honeynet because it takes a few steps to copy the data back against creating a complete image of hard drive and rolling back for real system. Virtual honeynet also offers the advantage of backing up the compromised system for later analysis [7].

III. Dynamic Honeypots

Honeypots like all network security technology needs configuration. Traditionally whether low, medium or high interaction of honeypots, they normally configure manually. The current static honeypot faces the same requirements. Even if configuration and deploying was not sufficient, we need someone to maintain the honeypot and it can be more challengeable. Due to solve this problem, the idea of the Dynamic Honeypots comes out. Possibly it seems a plug and play solution. We can simply plug it into the network and the honeypot does all the work for us [8].

Dynamics honeypots are all in demands including how many honeypots need to deploy, how to deploy them, and what they should look like will automatically determines so that they can blend in with the organization environment. Even better, the deployed honeypots can change and adapt to the organization environment. For instance, when we add Linux to the network, we rapidly have Linux honeypots or if we remove a Novell server from organization network, the Novell honeypots automatically disappear. When we change the router in our network, honeypot routers will change as well. The final goal is an application and a solution where we just simply plug into our network, then it will learn the environment, deploys the honeypots and capture the damages done by the attacker.

IV. Dynamic Honeypots Reviewed Algorithms

A. Algorithm 1

The article [9] has shown an independent honeypot capable of adapting in a dynamic and constantly changing network environment .In fact, they response to a proposed idea by honeynet organization. Thus, the dynamic honeypot appeal to integrate passive or active probing and virtual honeypots. Their propose design overview was the component which listed below:

1) An active probing tools to learn network such as Nmap [10].

2) A passive fingerprinting tools to learn network ,such as Passive Operation Finger Printing (Pof) [11].

3) A low interaction honeypot.

4) Physical honeypot to receive redirect traffic, such as Sebek.

5) A Database consist of hosts description and Log files.

Their implementation was combination of low and high interaction honeypot. They used Nmap and passive fingerprinting (P0f) to learn about network and they also use snort which is the network intrusion detection system that performs real-time traffic analysis and detects a variety of attacks and probes, such as buffer overflows, stealth port scans, and OS fingerprinting attempts. Honeyd used as a low interaction honeypots and Sebek as a high interaction honeypots. A most critical advantage of using this method to approach of dynamic honeypot trick in its capability to capture malicious attacks both on the small scale i.e. from host perspective (physical honeypot logs) and the large scale i.e. from the network perspective (virtual honeypot logs).

B. Algorithm 2

Researchers in the article [12], clarify a method to an automatically and dynamically configure and construct honeypots in consequences of the network scanning results. In addition to it, by collecting information the administrator will be capable to protect network in superior way .Their purpose of this research is to produce one product which system administrator or researcher can swiftly construct a honeypot without the need of professional knowledge of the honeypot configuration mechanism. Moreover, the honeynet administrator is not obligatory to identify the details of the network topology or installed systems in the network to be mimicked. Furthermore, simplify configuration and reconfigured to reproduce the current network.

The specification to approach this phenomenon is considered as follows. For manager tools and identified network Honeyd [14] in Perl and Linux OS environment was used as part of this project which .Then, Nmap for operation system detection and to determine which TCP/UDP ports which are open. To construct the configuration file for honeypot or honeynet, active scanning quickly provides sufficient data while it does consume some bandwidth. The Nmap results are analyzed real-time by the honeyd Configuration Manager to establish the resulting honeynet configuration, which is stored as a honeyd configuration file. Honeyd use the IP assignment method to create the file which administrator use to determine the network configuration of the honeypots. These IP assignments consent to the administrator to produce replicas of the scanned machines. The four options for IP address assignment are as follows:

First, configure the honeypots to use the same IP addresses as the real systems. Second, adjust the network component of the IP address, while preserving the host component. Third, use a selected IP address range in which to place the ensuing honeypots. Forth, mingle the honeypot systems into the construction network where possible, although this requires that the assigned IP addresses on a given production network can be identified. In addition, to

initiate a configuration which emulates the production hosts and open ports, a small MySQL database was used to accumulate the details about the obtainable scripts, including which services they emulate, and for what operating systems they are suitable. Thereby, when an open port is found by the Honeyd Configuration Manager is possible to quickly query the database for an appropriate script.

C. Algorithm 3

Honeyd@WEB [13] is low-interaction, production, dynamic and manageable virtual honeypots via a web interface. Base on the placement of honeypot, may be set up in front of a firewall, in the Demilitarized Zone (DMZ), or behind a firewall. However, this research recommended that the best place to set up a honeypot is behind a firewall. Consequently it is competent to detect internal attackers. In addition, the earlier the honeypot is to actual servers (which are probable behind a firewall), the more expected it is to attract intruders. Practically, the low-interaction honeypot (honeyd) was integrated with P0f to develop a dynamic and manageable honeypot. Honeyd@WEB using a web-based approach to sets and deploys the honeypots. It runs P0f, Arpd (Address Resolution Protocol Daemon) and honeyd through the web interface. The aim of using P0F is listening to traffic of the production network and determine the hosts ‟ operating systems by passive fingerprinting technique''. Regarding to information which are gathering by P0f such as the number of active hosts and their operation system it will suggest the deployment of honeyd.

Using of ARP is to map Internet Protocol address (IP address) to a physical machine address that is identified in the local network. An Arp daemon which is known as Arpd listens on a particular interface and answers ARP requests for some preferred IP addresses. Here, Arpd is used to direct network traffic towards the honeypot and to get honeyd to respond to all unused IP addresses on the network.

To deploying, editing or deleting administrator can then use the web interface honeypots. For logging and analysis purposes, Snort and ACID are used to detect attacks .But we should notice the amount of captured data is limited to probes and connection attempts since low interaction honeypots are used.

D. Algorithm 4

[14] Proposed a methodology in Linux base host by having a single physical network interface card. The goal of this research was to establish a virtual honeynet on a VMware Server running Honeywall CDROM [15] then, using data capture mechanisms reports help administrator for risk evaluation. According to the results of this study administrator is capable to enhance the overall security of our network resources.

This project was based on free and open source tools so Linux base operation system have been used the host OS (OS virtual machine). Likewise, the particular honeypot that integrated with Linux was Honeywall. Honeywall CDROM is a bootable CD that setups onto a hard drive and comes with all the tools and functionality for you to implement data capture, control and analysis. Moreover, VMware Server was used as the virtualization solution for this project due to free, reliable and an enormous proportion of support. Furthermore, Sebek High interaction honeypot was sets as a printer driver name.

During the implementation author identify one problem regarding the loop in the topology and the honeypot LAN segment was causing due to configuration in eth0 and eth1 interfaces which were as a VMware bridge and eth2 as a Vmware host-only interface, but they overcome this problem by reference the Pakistanian solution research [16]. They suggest Vmnet0 on the etho as a VMware bridge interface in the direction of router. VMnet 1 on the eth1 is a VMware host only interface leading to the internal LAN segment where honeypot is there and VMnet2 on the eth2 as a VMware bridge interface which is firewalled and accessible for remote management purpose. In additional, the public IP assign to the management interface within the honeynet subnet but it has a limited access to it from ROO configuration. Finally, this virtual honeynet was online for estimate 60 days from the attack results were documented as attacked ports and services, attacker IP's and country of origin.

However, the method which this researcher offered required a great amount of memory for virtualized environments and can be used as a performance benchmark. Similarly, using tools and technology such as Linux, Vmware and Snort demand a high degree of skill and customization. Not only installing, maintenance needs understanding of the system with deep knowledge regarding the network topology, nevertheless requires several software to approach this method.

V. Result and Analyzie

Studies indicate integration of networking tools and applications is a fundamental issue to approach easy deployment of honeypot. Table I illustrates the summary of methods regarding the researcher's algorithms.

TABLE I :Comparitive Table on Dynamic Honeypots

Research Title	Method	Comment
A dynamic honeypot design for intrusion detection	Combination of low and high interaction honeypots (Honeyd,Nmap,P0f,Sebek).	Complicated ,Buffer overflow,Using different software,Linux OS,Difficult to configuration.
Dynamic honeypot construction	Low interaction honeypot and Active finger printing (Nmap).	Consume a lot of bandwidth,Linux OS.
Honeyd@Web	Low interaction honeypot and Passive finger printing (honeyd,P0f,ARPD,Snort).	Limit of dynamic property,Inside firewall , Linux OS,Needs user for maintenance.
Experiences with a generation III virtual honeynet	Honeywall with High interaction honeypot (Sebek),Vmware,Snort,TCp dump,H flow P0f.	Linux OS,Utilizing Large amount of memory, Demand high degree of knowledge for using and customizable of tools,Needs a lot of maintenance.

VI. Conclusion

The value of honeypot is obvious in the security. Despite; deploying a honeypot was the most controversial concerns in this era. To deal with this problem researcher propose solution for easy deployment of dynamic honeypot. In this paper we consider four different solutions which is suggested to deal with installing and configuration problem .Then we compare them and account their advantages based on the easy deployment aims.

Acknowledgment

We would like to express our sincere gratitude toDR.MOHAMMAD ZIAUR RAHMAN and ARASH HABIBI LASHKARI for their supervision and guidance. Also, we would like toexpress our appreciation to our parents and all theteachers and lecturers who help us to understand theimportance of knowledge and show us the best way togain it.

References

[1] K. Scarfone and P. Mell, "Guide to Intrusion Detection and Prevention Systems (IDPS)," U. S. D. o. Commerce, Ed, ed. *Gaithersburg: lInstitute of Standards and Technology*, 2007.

[2] M. Yuqing, et al, "J-Honeypot: a Java-based network deception tool with monitoring and intrusion detection," in Information Technology: Coding and Computing, 2004. in *Proc. ITCC* 2004. International Conference on, 2004, pp. 804-808 Vol.1.

[3] G. M. Bednarski and J. Branson. (2004, 2010). Understanding Network Threats through Honeypot Deployment.[Information Warfare:].Available:http://www.infinitel00p.com/library/honeypot.pdf

[4] K. STEDING-JESSEN, et al, "Using Low-Interaction Honeypots to Study the abuse of Open Proxies to Send Spam," ed: Brazilian Network Information Center - NIC.br Computer Emergency Response Team Brazil - CERT.br, São Paulo (SP), 2007, p. 9.

[5] G. Wicherski, "Collecting and Managing Accumulated Malware Automatically"presented at the IN-DEPTH SECURITY CONFERENCE EUROPE, Vienna, Austria 2007.

[6] A. A. Ashour, "Designing high interaction windows honeynets," M.S. 1437339, University of Louisville, *United States - Kentucky*, 2006.

[7] A. Lanoy and G. W. Romney, "A Virtual Honey Net as a Teaching Resource," in Information Technology Based Higher Education and Training, 2006. ITHET '06. 7th *International Conference on*, 2006, pp. 666-669.

[8] R. Budiarto, et al, "Honeypots: why we need a dynamics honeypots," in Information and Communication Technologies: From Theory to Applications, 2004. Proceedings. 2004 I*nternational Conference on*, 2004, pp. 565-566.

[9] I. Kuwatly, et al, "A dynamic honeypot design for intrusion detection," in Pervasive Services, 2004. ICPS 2004. Proceedings. *The IEEE/ACS International Conference on*, 2004, pp. 95-104.

[10] Nmap, "Nmap Security Scanner 5.20 ed: Gordon Lyon, 1997, p. Used to discover hosts and services on a computer network.

[11] M. Zalewski, " Passive OS fingerprinting tool," 2000-2006, Ed, 2 ed: *Michal Zalewski*, 2000.

[12] C. Hecker, et al, "Dynamic Honeypot Construction," in *Proc. of the 10th Colloquium for Information Systems Security Education*, University of Maryland, University College ,Adelphi, MD, 2006.

[13] HONEYD,"Developments of the Honeyd Virtual Honeypot, "Honeyd1.5 c ed: Monkey.org, 2007.

[14] F. H. Abbasi and R. J. Harris, "Experiences with a Generation III virtual Honeynet," in *Telecommunication Networks and Applications Conference (ATNAC)*, 2009 Australasian, 2009, pp. 1-6.

[15] Honeynetproject. (2005, 1st November). Know Your Enemy:Honeywall CDROM Roo 3rd Generation Technology.

[16] F. A. Shuja. (2006, 1st November). V*irtual Honeynet: Deploying Honeywall using VMware.*

Hamid Mohammadzadeh.e.n becomes a Member (M) of IACSIT. Hamid was born in Mashhad, Iran on 3rd August 1979. Bachelor degree in software engineering from university of Applied Science and Technology , Mashhad, Iran in 2006 and master degree in computer science from University of Malaya, Kuala Lumpur, Malaysia in 2011.

Hamid was researching on the dynamic honeypots in the University of Malaya in the network security specializing lab. The related publications are Taxonomy on Hybrid honeypots and Comparison of Fingerprinting Techniques in Deployment of Dynamic Honeypots. Currently works as a research team leader in the Mahab Information Technology Company in the Khorassan Technology Park, Mashhad, Iran and the main focus of these researches are graphical password and deployment of dynamic honeypots. Interests' researches are client honeypots, Dynamic Honeypots, Honeypots against Botnet and network security tools.